LZT 108 3339

Mobile Phone R 380 Design Guidelines for WAP Services

ERICSSON

Ericsson is the leading provider in the new telecoms world, with communications solutions that combine telecom and datacom technologies with freedom of mobility for the user. With more than 100,000 employees in 140 countries, Ericsson simplifies communications for its customers - network operators, service providers, enterprises and consumers - the world over.

WAP™ and WAP Forum™ are registered trademarks of the WAP Forum, LTD., a registered company in the United Kingdom.

Second edition (June 2000)

This document is published by **Ericsson Mobile Communications AB**, without any warranty. Improvements and changes to this text necessitated by typographical errors, inaccuracies of current information or improvements to programs and/or equipment, may be made by **Ericsson Mobile Communications AB** at any time and without notice. Such changes will, however, be incorporated into new editions of this document.

All rights reserved.

© Ericsson Mobile Communications AB, 2000

Contents

PREFACE 2

Who Should Read this Guide? 2

Overview of Contents 2

Related Publications 2

Ericsson Publications 2

Other Publications 3

Terminology 3

Abbreviations 3

Definitions 3

GENERAL USABILITY ASPECTS 5

USER INTERFACE ISSUES 6

The Browser Display 6

The On-screen Keyboard 7

Navigating 7

DESIGN COMPONENTS 8

Using Text 8

Font Size and Font Style 8

Line Spacing and Line Breaks 9

Paragraphs 10

Indented Paragraphs 11

The Card Title Bar 12

Using Select Lists 12

Single Choice Lists 12

Multiple Choice Lists 13

Using Buttons 14

Using Input Fields 15

Using Images 17

Using Hyperlinks 19

Using Images as Hyperlinks 20

Using Tables 21

THE PIZZA PENGUIN EXAMPLE 24

PREFACE

This document describes how to design WAP services for the Ericsson Mobile Phone R380.

Who Should Read this Guide?

This document is intended for developers of WAP services who want a deeper insight into designing applications for the Ericsson R380 WAP browser.

It is assumed that the reader is familiar with the WAP concept and WML.

Overview of Contents

The chapters in this document are organized as follows:

General Usability Aspects

This chapter gives an overview of issues to be taken into account when developing WAP services for Ericsson R380.

User Interface Issues

This chapter gives a description of the physical and logical input/output devices of the R380 and the way these are used together with WAP services.

Design Components

This chapter contains information about graphical user interface components. The chapter also contains several examples of WML code as well as figures showing how these are presented in the browser.

The Pizza Penguin Example

This chapter contains a complete WAP application in WML code.

Related Publications

Ericsson Publications

[1] *Ericsson WAP Info*
http://www.ericsson.com/WAP/

[2] *WAP White Paper*
February 1999
AU-System AB
http://www.ericsson.com/WAP/what/wapwp2.pdf

Other Publications

[3] *Wireless Application Protocol Architecture Specification*
Version 30-Apr-1998
Wireless Application Protocol Forum, Ltd.
http://www.wapforum.org/what/technical.htm

[4] *WAP Wireless Markup Language Specification, Version 1.1*
Version 16-JUN-1999
Wireless Application Protocol Forum, Ltd.
http://www.wapforum.org/what/technical.htm

Terminology

Abbreviations

DTMF	Dual Tone Multi-Frequency
GIF	Graphics Interchange Format
HTTP	Hypertext Transfer Protocol
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
WAP™	Wireless Application Protocol
WBMP	WAP Bitmap
WML	Wireless Markup Language

Definitions

Bookmark	A named reference to a URL. Enables the user to keep track of favorite pages and then return to them, easily, later on.
Card	A single WML unit of navigation and user interface. May contain information to present to the user, instructions for gathering user input, etc.
Card Contents	Subject data stored or generated at an origin server. The card contents display text, images, buttons and other controls supported by WML as a response to a user request. When a user performs an action in the card contents, the WAP Browser responds according to the WML specification.
Card Title	An attribute in the WML Card element specifying advisory information about the card.
Card Title Sequence	A generic concept for visualizing the progress and history of a WML application. The number of titles in the sequence is dependent on the number of cards that have been browsed.
Current Card	The card whose contents is currently displayed in the browser.
Deck	A collection of WML cards.
Image	WBMP or GIF image contained in a Card.

newcontext	An attribute in the Card element indicating that the current browser context should be re-initialized upon entry to this card. The re-initialization includes removal of all previously defined variables and a reset of the navigation history.
WAP Application	A collection of WML cards, with the newcontext attribute set in the entry card.
WAP service	A WML application residing on a web site.
WAP Services	The R380 WAP browser.

GENERAL USABILITY ASPECTS

When it comes to designing any kind of electronic service, simplicity is the key to gaining acceptance and appreciation from users. Simplicity is even more important in a mobile environment since the user often needs to concentrate on other things as well as using the terminal.

Keep information and text short and meaningful. Since overview is crucial, keep lists and menus fairly short. Do not force the user to scroll through more than 4-5 displays (corresponds to 25-35 lines). Try to avoid long words and abbreviations because the screen resources are limited. Use short and meaningful words instead of hard to understand abbreviations that might be misinterpreted.

Remember that there is a size limit to what can, in practice; be sent to the terminal in one deck. Keep the size of the encoded decks at less than 3.800 bytes (images not included). Images should not be more than 3.800 bytes per image and the number of images per card not more than 30. Use images to supplement text rather than replace it. Depending on service, it could be wise to stay well below the above-mentioned limits from a usability perspective. We recommend user testing on all services being done in a live environment.

Use few levels in menu hierarchies since:

1. The user can easily get lost in complex and deep menu systems.
2. Depending on the bearer and the structure of the WML code, it could be slow to move down through a multi-level menu system.

Use header texts since headers are used as card titles, which describe the contents of a card. Card titles should say something about the context the user is currently in. Card titles are also a navigational aid for users, helping them to remember where they are in an application.

Use links to access more information. Links are a powerful tool for linking different information items and preventing long pages of text.

Tip: *An R380 browser can be identified at the server side by the contents of the HTTP/1.1 header User-Agent request field. At the server side knowledge of the identity of the requesting browser then can be used in order to start WAP services tailor-made for this particular browser.*

The above-mentioned identity of the R380 browser is:
“R380 2.0 WAP1.1”

Unfortunately no standardized method exist in order to pull out this information. The methods are different depending brand and version of the Internet server used. Please consult your Internet server documentation for details on how to make use of the information contained in the HTTP header fields.

USER INTERFACE ISSUES

The Browser Display

The Ericsson R380 display is a grey scale black and white touchscreen with a resolution of 360x120 pixels with 0.23 pitch. This means an active screen size of almost 83x28mm.

The browser display has three sections: the Browser Area, the Card Title bar at the top, and the Toolbar to the right.

Browser Area

The Browser area is where the card content is presented to the user. The size of the area is 310x100 pixels. This includes a top margin of 2 pixels and a left and right margin of 3 pixels each, which reduces the size of the usable area.

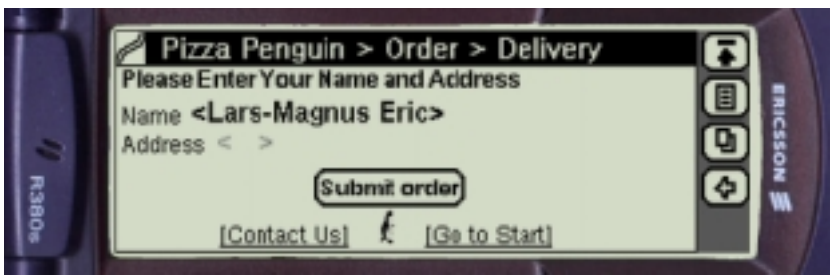
If a card content is too extensive to be visible in full, only the beginning of the card will be visible when first displayed. In this case a vertical scroll bar appears to the right in the browser area. The scroll bar reduces the browser area width by 16 pixels.

Graphical components, text and images are displayed in the browser area starting in the upper left-hand corner and shown in the same order as they appear in the WML code.

Card Title Bar

The Title bar shows which card is presently being displayed and also the cards that have been browsed through previously. If there are more cards being browsed through than can fit into the Title bar, only the most recent titles are displayed. This is a good reason to keep card titles short and meaningful.

This navigation history is reset when a Card with the `newcontext` attribute set to *true* is reached.



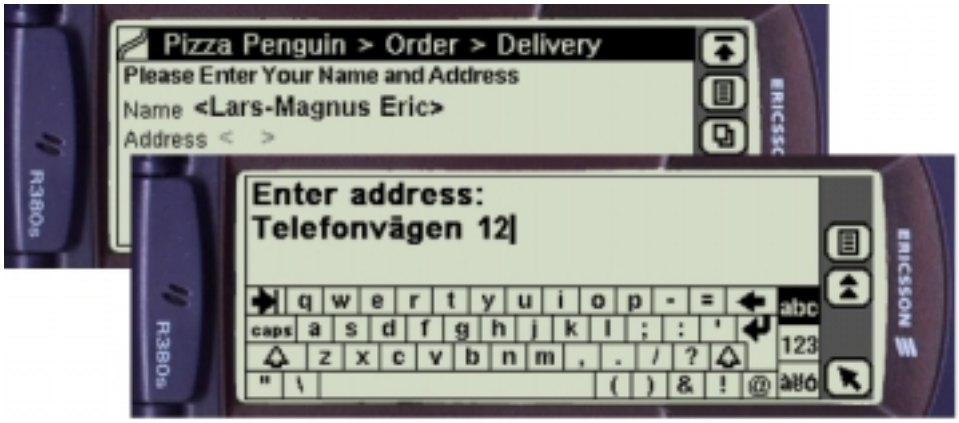
Toolbar

The Toolbar is an area containing buttons required by the browser.

The On-screen Keyboard

Text input is done with on-screen keyboards or a character recognition screen. Three different keyboard layouts are available: an alpha layout, a numeric layout and a layout with national characters.

Touching anywhere in an input field makes the keyboard, or character recognition screen, appear.



Navigating

In order to navigate to a WAP service, three methods are available.

- Frequently visited locations can be bookmarked once and then reached easily using the Bookmark. When bookmarking, the title of the application entry card is the default name of the bookmark. The number of bookmarks that can be saved is in practice almost unlimited.
- Newly visited locations can be reached by using the History list.
- New locations can be reached by entering the URL in the Open Location dialog using the on-screen keyboard.

Browsing through information in an application is done by activating hyperlinks and touching buttons in the browser area.

In order to provide an easy way of navigating backwards, the Toolbar contains a Back button.

Touching the Back button navigates the user back in the WAP services. Next after the first card in the current service (defined by having the the `newcontext` attribute set to *true*) has been reached, the first card in the previous service is displayed.

DESIGN COMPONENTS

This chapter gives an overview of the graphical user interface components and a description of their corresponding WML elements. These items are then used when designing WAP services for the R380.

The “Look and Feel” of the components is described and guidelines are made on component usage. The WML elements and their attributes are also listed and a brief description is given.

The WML element descriptions are, supplemented with WML code as well as figures showing the way this is displayed in the browser area of the R380.

Note: *The description below does not include all possible WML elements and attributes, but all elements affecting the graphical user interface.*

Using Text

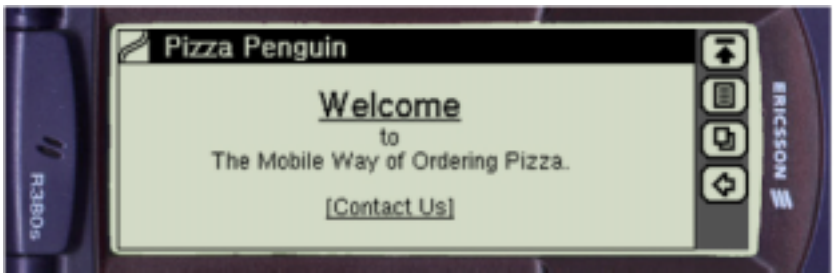
Font Size and Font Style

The font used in R380 is a proportional font. A proportional font gives a dynamic and minimized width for each character, which improves readability of the displayed texts. This generally allows more characters to be displayed per line.

The used typeface name is “Swiss A” and three different font sizes are used: small, normal and big.

Font size	Height	Number of lines
Small	9 pixels	7.5
Normal	10 pixels	7
Big	12 pixels	5.5

The normal font can be shown in either regular or bold style, the small and big fonts only in bold style.



WML

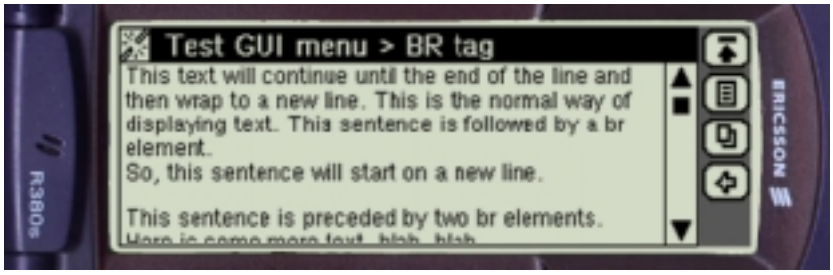
The R380 browser supports all emphasis elements. The `big` and `small` elements change the font size. The `em`, `strong`, `i`, `b` and `u` elements are all handled in the same manner (all fonts are shown in bold).

A text formatting example

```
<wml>
<card id="first" title="Pizza Penguin" newcontext="true">
<p align="center">
   <br/>
   <b><big><a href="#second">Welcome</a></big></b><br/>
   to<br/>
   The Mobile Way of Ordering Pizza.<br/><br/>
   <a href="#fifth">[Contact Us]</a>
</p>
</wml>
```

Line Spacing and Line Breaks

The line spacing in the R380 browser is single line with one additional pixel before the line and one after. If a line is too long to fit horizontally in the browser area, the line will be word-wrapped onto multiple lines. Any suitable inter-word space is used as line break point.


Test GUI menu > BR tag
This text will continue until the end of the line and then wrap to a new line. This is the normal way of displaying text. This sentence is followed by a br element.
So, this sentence will start on a new line.
This sentence is preceded by two br elements.
R380s
ERICSSON

Text can co-exist with images, select lists, buttons, input fields and hyperlinks all on the same line.

WML

The beginning of a new line is defined by a `br` element.

The `br` element affects all contents in the browser view. Text, images, select lists, buttons, input fields, hyperlinks and tables are positioned on a new line according to the current alignment.

The `br` element can also be used inside a table cell (see Using Tables below).

A line break example

The example below shows how the `br` element will affect the card content appearance.

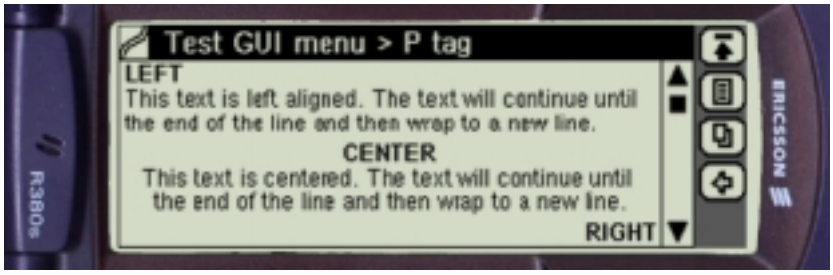
```
<wml>
<card id="init" title="BR tag">
<p>
   This text will continue until the end of the line and
   then wrap to a new line. This is the normal way of
   displaying text. This sentence is followed by a br
   element.<br/>
   So, this sentence will start on a new line.<br/><br/>
   This sentence is preceded by two br elements.
   Here is some  more text, blah, blah, ...
</p>
</card>
</wml>
```

Paragraphs

Text on a WML card comprises one or more paragraphs.

A paragraph always starts on a new line and is preceded by an extra line space of 3 pixels.

Text within a paragraph is by default displayed left aligned. However, it can optionally be displayed right aligned or centered.


Test GUI menu > P tag
LEFT
This text is left aligned. The text will continue until the end of the line and then wrap to a new line.
CENTER
This text is centered. The text will continue until the end of the line and then wrap to a new line.
RIGHT
R380s
ERICSSON

WML

Paragraphs are defined by using the p element which also establishes the alignment attribute for a paragraph.

Attribute	Description
align	The align attribute can take the following values: *left*, *right* and *center*. Left alignment is the default alignment mode.
mode	The mode attribute specifies the line-wrap mode for the paragraph and can take the following values: *wrap*, which specifies breaking text mode and *nowrap*, which specifies non-breaking text mode. Breaking text mode is the default mode. In non-breaking text mode, a horizontal scroll bar appears if any line in the WML card is to long to fit into the browser area.

A paragraph example

The example below shows how the p element and the align attribute affect the card content appearance.

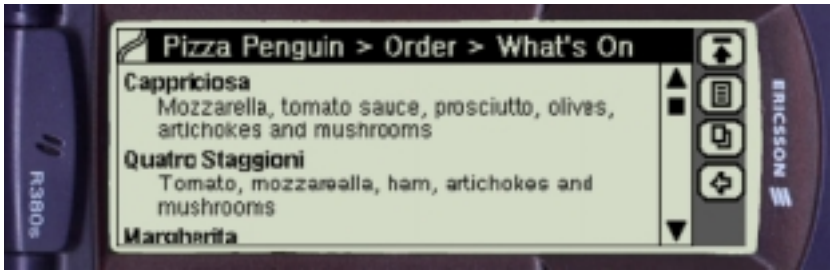
```
<wml>
<card id="init" title="P tag">
<p>
   <b>LEFT</b><br/>
   This text is left aligned. The text will continue until
   the end of the line and then wrap to a new line.<br/>
</p>
<p align="center">
   <b>CENTER</b><br/>
   This text is centered. The text will continue until the
   end of the line and then wrap to a new line.<br/>
</p>
<p align="right">
   <b>RIGHT</b><br/>
   This text is right aligned. The text will continue until
   the end of the line and then wrap to a new line.
</p>
</card>
</wml>
```

Indented Paragraphs

Related text and other components can be grouped together to increase usability. Such groups can be nested.

Each group is indented 20 additional pixels and preceded by an extra line space of 3 pixels.

A header in bold font can optionally precede a group. In such a case a so-called hanging indent applies.



WML

The beginning of a group is defined by using the `fieldset` element.

Attribute	Description
`title`	The value of the `title` attribute, if present, will be used as a leading text to the group of text and objects. The text following the title will be located on a new line.

A `fieldset` example

The example below shows how the use of `fieldset` elements affects the card content appearance.

```
<card id="pizzainfo" title="What's On">

<p>
   <fieldset title="Cappriciosa">
      Mozzarella, tomato sauce, prosciutto, olives, artichokes
      and mushrooms
   </fieldset>

   <fieldset title="Quatro Staggioni">
      Tomato, mozzarella, ham, artichokes and mushrooms
   </fieldset>

   <fieldset title="Margherita">
      Fresh tomatoes, fresh mozzarella &amp;amp; basil
   </fieldset>
   .
   .
   .
   .
</p>
</card>
</wml>
```

The Card Title Bar

See Card Title Bar on page 6.

WML

The title in the Card Title bar is defined by using the `title` attribute in the `card` element.

Using Select Lists

A select list is an input component that specifies a list of options for the user to choose from. Single and multiple choice lists are supported.

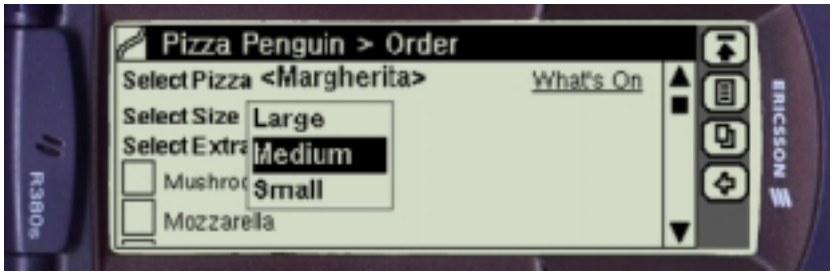
Single Choice Lists

A single choice list is used for the selection of one and only one item in a list. The list is presented as a drop-down listbox.

In its closed state the listbox displays the currently selected value within angled brackets (see Select Pizza in the example below). The component is 15 pixels high and surrounded by 5 pixels of white space to the left and right. The width adapts to the length of the text within the brackets. However, if the total width should have exceeded 150 pixels the text is truncated.

In its opened state, a list with an optional scrollbar is displayed. The number of lines is unlimited (see Select Size in the example below).

The text in a single choice list is displayed using the Normal bold font.



WML

Two elements define a single choice list: `select` and `option`.

Single choice lists are specified using the `select` element with the `multiple` attribute set to *No*. Each list item is specified by an `option` element.

The `select` element

Attribute	Description
`multiple`	The `multiple` attribute can take the following values: *Yes* and *No*. The default value is *No*. To create a single choice list, this attribute is set to *No*.

The `title` attribute is not supported.

The `option` element

Text in the `option` element content is used in the drop-down list.

The `title` attribute is not supported.

An example of a single choice list

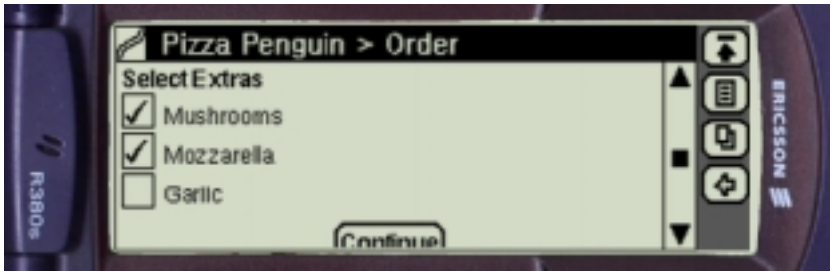
```
<p>
   <b>Select Pizza</b>
   <select>
      <option>Cappriciosa</option>
      <option>Quatro Staggioni</option>
      <option>Margherita</option>
      <option>Campagnola</option>
      <option>Siciliana</option>
   </select>
   <a href="#pizzainfo">What's On</a>
</p>

<p>
   <b>Select Size</b>
   <select>
      <option>Large</option>
      <option>Medium</option>
      <option>Small</option>
   </select>
</p>
```

Multiple Choice Lists

A multiple choice list is used for the selection of zero, one or more items in a list. The list is presented as a number of check boxes, one for each choice.

Each check box is located on a new line and followed by 2 pixels of white space.



Multiple choice lists may be organized into hierarchical groups. Each new hierarchical group is indented 20 pixels and preceded by an extra line space of 3 pixels. A header in Normal regular font can, optionally, precede a group.

WML

Three elements define a multiple choice list: `select`, `option` and `optgroup`.

Multiple choice lists are specified using the `select` element with the `multiple` attribute set to *Yes*. Each check box item is specified by an `option` element.

The `select` element

Attribute	Description
`multiple`	The `multiple` attribute can take the following values: *Yes* and *No*. The default value is *No*. To create a multiple choice list, this attribute is set to *Yes*.

The `title` attribute is not supported.

The `option` element

Text in the `option` element content is displayed to the right of the check box using the Normal regular font.

The `title` attribute is not supported.

The `optgroup` element

The `optgroup` element allows grouping of related `option` elements into a hierarchy.

Attribute	Description
`title`	The value of the `title` attribute is used as a leading text to the group of options. The text is located on a new line.

An example of a multiple choice list

```
<p>
   <b>Select Extras</b>
   <select multiple="true">
      <option>Mushrooms</option>
      <option>Mozzarella</option>
      <option>Garlic</option>
   </select>
</p>
```

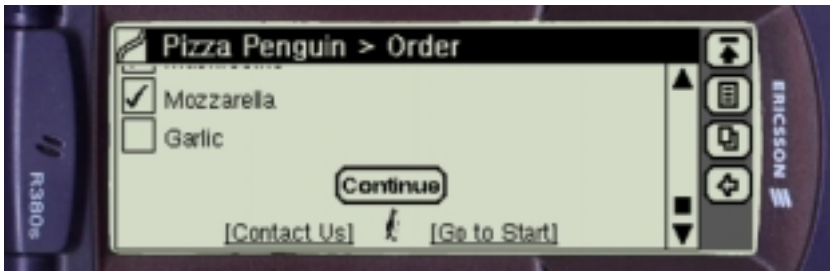
Using Buttons

Buttons are components that start actions or change states.

Buttons are shown as rounded rectangles and designed to contain text of variable length. Buttons have a constant height and a minimum, standard width, from which it can be lengthened to accommodate the contained text. The text in a button is displayed using the Normal Bold font.

Buttons are placed seven pixels below the current baseline and surrounded by 5 pixels of white space to the left and right and 1 pixel above and below.

A button may be located anywhere in the text flow of the browser area.



WML

Buttons are defined by using the `do` element.

Pressing the button invokes the action defined in the `do` element.

The R380 supports the following attributes:

Attribute	Description
`label`	The value of the `label` attribute is used as text on the button. The text is not truncated.
`type`	If no `label` is specified, then the value of the `type` attribute is used on the button. The `type` attribute can take the following values: *accept, prev, help, reset, options, delete* and *unknown.*
`name`	The value of the `name` attribute specifies the name of the `do` event binding. If `name` is not specified it defaults to the value of the `type` attribute.

Note: *If two do elements have identical type values, different name attribute values must be specified in order to create unique bindings. If the bindings are not unique, only the first button will be shown.*

A `do` example

```
<p align="center">
   <do type="accept" label="Continue">
   <go href="#third"/>
   </do><br/>
</p>
```

An example of three `do` elements of the same type

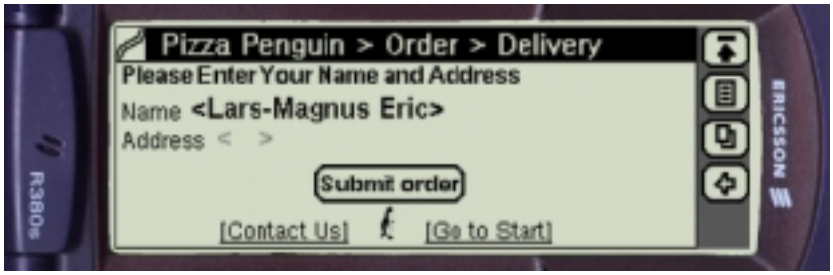
```
<p align="center">
   <do type="accept" label="Continue" name="cont">
   <go href="#third"/>
   </do><br/>
   <do type="accept" label="Contact Us" name="contact">
   <go href="#fifth"/>
   </do>
   <do type="accept" label="Go to Start" name="start">
   <go href="#first"/>
   </do>
</p>
```

Using Input Fields

Input fields are the only components enabling a user to enter text.

The input field displays the text within angled brackets. The component is 15 pixels high and surrounded by 5 pixels of white space to the left and right. The width adapts to the length of the text within the brackets. However, if the total width should have exceeded 150 pixels the displayed text is truncated. An empty field is displayed as if it contained three space characters.

The text in an input field is displayed using the Normal bold font.



Touching the screen anywhere in an input field makes the keyboard, or character recognition screen, appear (see The On-screen Keyboard above).

The number of characters a user can enter is in practice unlimited.

WML

Input fields are defined by using the `input` element.

The following attributes are supported:

Attribute	Description
`type`	The `type` attribute can take the following values: *text* and *password*. If `type` is set to *password*, the characters entered in the On-screen Keyboard are replaced by asterisks when displayed.
`value`	If no pre-load value is defined for the input object, the value of the `value` attribute is used as a default text in the input field.
`format`	Specifies an input mask for user input entries. The string consists of mask control characters and static text that is displayed in the input area. The format control characters specify the data format expected to be entered by the user. The format codes are: **`A`** Upper case alphabetic or punctuation, **`a`** Lower case alphabetic or punctuation, **`N`** Digit, **`X`** Upper case alphabetic character, **`x`** Lower case alphabetic character, **`M`** Any character, **`m`** Any character, **`*f`** Any number of characters of type f, **`nf`** n number of characters of type f, **`\c`** Display the special character c, in the entry field.

`maxlength`	Maximum number of characters a user can enter.
`title`	The value of the `title` attribute is used as label in the first line of the On-screen Keyboard component.

An input example

```
<p>
   Name<input type="text" title="Enter name:" name="name"/><br/>
   Address<input type="text" title="Enter address:"
   name="address"/>
</p>
```

Using Images

The R380 browser supports images in WAP bitmap (WBMP) and GIF formats.

WBMP supports black and white.

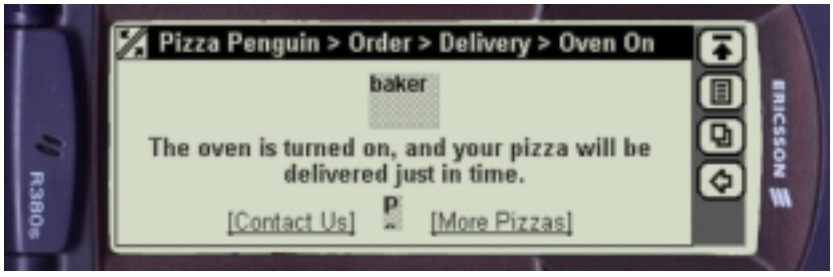
If the GIF format is used, four different “colors” are available:

White	0% black
Light-grey	25% black
Mid-grey	50% black
Black	100% black

Pizza Penguin > Order > Delivery > Oven On
The oven is turned on, and your pizza will be delivered just in time.
[Contact Us]
[More Pizzas]
R380s
ERICSSON

An image is always loaded after the WML code. During the load time, the browser displays a rectangular placeholder. See the figure below.

However, the R380 user can choose not to load images at all. In this case, the placeholder is permanent.


Pizza Penguin > Order > Delivery > Oven On
baker
The oven is turned on, and your pizza will be delivered just in time.
[Contact Us]
[More Pizzas]
R380s
ERICSSON

There is no limitation on the height and width of an image. A user must scroll in order to view a large image.

Note: *The number of images should not exceed 30 per card.*

WML

The `img` element indicates that an image is to be included in the text flow.

The R380 supports the following attributes:

Attribute	Description
`alt`	An alternative textual representation for the image used in the placeholder.
`src`	The source (URI) of the image.
`vspace`	This attribute specifies the amount of white space to be inserted above and below the image. The default value is 0.
`hspace`	This attribute specifies the amount of white space to be inserted to the left and right of the image. The default value is 0.
`height`	This attribute indicates the vertical size of an image and is used to reserve space when rendering the card before the image is loaded.
`width`	This attribute indicates the horizontal size of an image and is used to reserve space when rendering the card before the image is loaded.
`align`	The `align` attribute can take the following values: *top*, *middle* and *bottom*. Bottom alignment is the default alignment mode for images. The value of the `align` attribute indicates what part of the image should be aligned to the corresponding part of the current text line.

The `vspace`, `hspace`, `height` and `width` attributes are only supported when specified as absolute values.

Tip: *Always include the height and width attributes when using images. This guarantees that the card layout pattern will be correct when first displayed, even if no images have yet been loaded.*

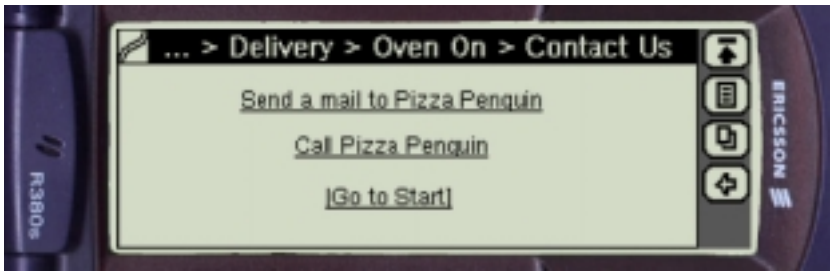
An image example

```
<p align="center">
   <img alt="baker"src="baker.gif" vspace="5" width="40"
   height="30"/><br/>
   <b>The oven is turned on, and your pizza will be delivered
   just in time.</b><br/>
</p>
```

Using Hyperlinks

Hyperlinks are underlined text or a graphic that can be tapped in order to:

- Navigate within a deck.
- Navigate to a new deck.
- Initiate a voice call.
- Send DTMF tones.
- Start composing an e-mail.
- Download an electronic business card (vCard).
- Download calendar data (vCalendar).



A hyperlink may be located anywhere in the text flow. Text in hyperlinks is displayed in the same font as the text preceding the link, but the text is underlined. Touching a hyperlink highlights the link text using reverse video. Touch and release on a link invokes the action associated to the link.

WML

Hyperlinks are specified using the `a` element. Below are examples of functions in the R380 that can be activated through hyperlinks.

Download an electronic business card

```
<a href="LarsMagnusEricsson.vcf"> Download Lars-Magnus Ericssons Business Card </a>
```

The file must have a `.vcf` extension or the content type must be set to `text/x-vcard` at the server.

Download calendar data

```
<a href="CourseSchedule.vcs"> Download Course Schedule Data </a>
```

The file must have a `.vcs` extension or the content type must be set to `text/x-vcal` at the server.

Initiate a voice call

```
<a href="wtai://wp/mc;+464690510">Call Pizza Penguin</a>
```

Initiates a voice call to the specified number. The number is displayed to the user who also must acknowledge the call.

Send DTMF Tones

```
<a href="wtai://wp/sd;9">Next Message</a>
```

Sends the specified DTMF tone sequence (i.e. the digit 9 in the example above) through an active voice connection. A voice call must have been initiated before the DTMF tones can be sent.

A complete hyperlink example

Three different links are shown in this example: a mailto link, a voice call link and an interdeck link.

```
<card id="fifth" title="Contact Us">
   <p align="center">
      <br/>
      <a href="mailto:pizza@penguin.nu">
      Send a mail to Pizza Penguin</a><br/><br/>
      <a href="wtai://wp/mc;+464690510">Call Pizza Penguin
      </a><br/><br/>
   </p>
   <p align="center">
      <a href="#first">[Go to Start]</a>
   </p>
</card>
```

Using Images as Hyperlinks

An image can also serve as a link. By default, there are no differences in image layout if the image serves as a link or not. However, the R380 browser can be set to show image links with a frame of 1 black and 1 white pixel. This affects the card layout.

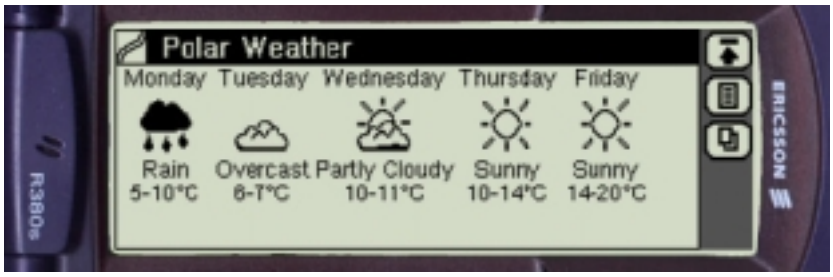
WML

An image hyperlink example

```
<p align="center">
   <br/>
   <a href="mailto:pizza@penguin.nu">
   <img alt="Penguin" src="mail.gif" width="44"
   height="50"/></a>
             
   <a href="wtai://wp/mc;+464690510"><img alt="Baker"
   src="phone.gif" width="42" height="50"/></a><br/>
</p>
```

Using Tables

Due to its wide display, the R380 browser has good support for tables. Tables are used to make information easier to grasp and analyze.

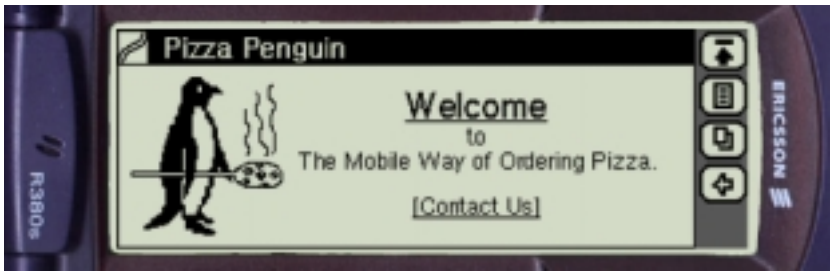


The R380 browser displays a table in columns and rows without a border. The table is located on a new line. Empty table cells and empty table rows are significant and are included in the presentation of the table.

A column's contents can be left aligned, centre aligned, or right aligned. The browser will handle a multiple-lines cell, which may result from wrapping of cell contents.

The width of each column is determined by the maximum width of the text or images in that column. If the width of the table turns out to be too large, the size of each column is reduced proportionally and the contents wrapped if possible.

Tables can also be used to format card contents that include images in an appealing way.



In addition to text and images, tables can also contain hyperlinks and images used as hyperlinks.

WML

The `table` element is used together with `tr` and `td` to create a set of aligned columns of text and images.

The `tr` element is used as a container to hold a single table row.

The `td` element is used as a container to hold a single table cell data.

Attribute	Description
`align`	Alignment of cell content is fully supported. The `align` attribute can take the following values: *L* (left), *R* (right) and *C* (centre). Left alignment is the default alignment mode.
`columns`	Specifies the number of columns for the table.

The Polar Weather `table` example

```
<card id="First" title="Polar Weather" newcontext="true">

<p>
   <table align="ccccc" columns="5">
   <tr>
      <td>Monday</td>
      <td>Tuesday</td>
      <td>Wednesday</td>
      <td>Thursday</td>
      <td>Friday</td>
   </tr>
   <tr>
   <td>
      <img alt="weather"src="rain.gif" width="30" height="30"
      hspace="10" vspace="3"/></td>
   <td>
      <img alt="weather"src="cloudy.gif" width="30" height="30"
      hspace="10" vspace="3"/></td>
   <td>
      <img alt="weather"src="sun_in_cloud.gif" width="30"
      height="30" hspace="10" vspace="3"/></td>
   <td>
      <img alt="weather"src="sun.gif" width="30" height="30"
      hspace="10" vspace="3"/></td>
   <td>
      <img alt="weather"src="sun.gif" width="30" height="30"
      hspace="10" vspace="3"/></td>
   </tr>
   <tr>
      <td>Rain</td>
      <td>Overcast</td>
      <td>Partly Cloudy</td>
      <td>Sunny</td>
      <td>Sunny</td>
   </tr>
   <tr>
      <td>5-10&#176;C</td>
      <td>6-7&#176;C</td>
      <td>10-11&#176;C</td>
      <td>10-14&#176;C</td>
      <td>14-20&#176;C</td>
   </tr>
   </table>
</p>
</card>
```

The Pizza Penguin `table` example

```
<card id="first" title="Pizza Penguin" newcontext="true">

<p>
   <table align="cc" columns="2">
   <tr>
   <td>
      <a href="#second"><img alt="Penguin" src="bigpenguin.gif"
      hspace="4" vspace="3" width="88" height="88"/></a></td>
   <td>
      <br/><b><big><a href="#second">Welcome</a></big></b><br/>
      to<br/>
      The Mobile Way of Ordering Pizza.<br/><br/>
      <a href="#fifth">[Contact Us]</a></td>
   </tr>
   </table>
</p>
</card>
```

THE PIZZA PENGUIN EXAMPLE

Below is a complete WML code listing from a WAP demonstration application. The examples in the previous chapters were mainly selected from this application.

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">

<wml>

<card id="first" title="Pizza Penguin" newcontext="true">
<onevent type="ontimer">
<go href="#second"/>
</onevent>
<timer value="70"/>

<p>
   <table align="cc" columns="2">
   <tr>
   <td>
      <a href="#second"><img alt="Penguin" src="bigpenguin.gif"
      hspace="4" vspace="3" width="88" height="88"/></a></td>
   <td>
      <br/><b><big><a href="#second">Welcome</a></big></b><br/>
      to<br/>
      The Mobile Way of Ordering Pizza.<br/><br/>
      <a href="#fifth">[Contact Us]</a></td>
   </tr>
   </table>
</p>
</card>


<card id="second" title="Order">

<p>
   <b>Select Pizza</b>
   <select title="Type">
      <option>Cappriciosa</option>
      <option>Quatro Staggioni</option>
      <option>Margherita</option>
      <option>Campagnola</option>
      <option>Siciliana</option>
   </select>
   <a href="#pizzainfo">What's On</a>
</p>

<p>
   <b>Select Size</b>
   <select title="Size">
      <option>Large</option>
      <option>Medium</option>
      <option>Small</option>
   </select>
</p>
```

```
<p>
   <b>Select Extras</b>
   <select title="Extras" multiple="true">
      <option>Mushrooms</option>
      <option>Mozzarella</option>
      <option>Garlic</option>
   </select>
</p>

<p align="center">
   <do type="accept" label="Continue">
   <go href="#third"/>
   </do><br/>
   <a href="#fifth">[Contact Us]</a>   
   <img alt="Penguin" src="xspenguin.gif" width="11"
   height="19"/>   
   <a href="#first">[Go to Start]</a>
</p>
</card>


<card id="third" title="Delivery">

<p>
   <b>Please Enter Your Name and Address</b>
</p>

<p>
   Name<input type="text" title="Enter name:" name="name"/><br/>
   Address<input type="text" title="Enter address:"
   name="address"/>
</p>

<p align="center">
   <do type="accept" label="Submit order">
   <go href="#sixth"/>
   </do><br/>
   <a href="#fifth">[Contact Us]</a>   
   <img alt="Penguin" src="xspenguin.gif" width="11"
   height="19"/>   
   <a href="#first">[Go to Start]</a> 
</p>
</card>


<card id="fifth" title="Contact Us">

<p align="center">
   <br/>
   <a href="mailto:pizza@penguin.nu">
   <img alt="Penguin" src="mail.gif" width="44"
   height="50"/></a>
             
   <a href="wtai://wp/mc;+464690510"><img alt="Baker"
   src="phone.gif" width="42" height="50"/></a><br/>
</p>

<p align="center">
   <a href="#first">[Go to Start]</a> 
```

```
</p>
</card>


<card id="sixth" title="Oven On">
<p align="center">
   <img alt="baker" src="baker.gif" vspace="5" width="40"
   height="30"/><br/>
   <b>The oven is turned on, and your pizza will be delivered
   just in time.</b><br/>
</p>

<p align="center">
   <a href="#fifth">[Contact Us]</a>   
   <img alt="Penguin" src="xspenguin.gif" width="11"
   height="19"/>   
   <a href="#first">[More Pizzas]</a> 
</p>
</card>


<card id="pizzainfo" title="What's On">

<p>
   <fieldset title="Cappriciosa">
      Mozzarella, tomato sauce, prosciutto, olives, artichokes
      and mushrooms
   </fieldset>

   <fieldset title="Quatro Staggioni">
      Tomato, mozzarella, ham, artichokes and mushrooms
   </fieldset>

   <fieldset title="Margherita">
      Fresh tomatoes, fresh mozzarella &amp; basil
   </fieldset>

   <fieldset title="Campagnola">
      Tomato, mozzarella, zucchini, eggplant, peppers and
      mushrooms
   </fieldset>

   <fieldset title="Siciliana">
      Carciofini, prosciutto, pomidoro, fresh mozzarella &amp;
      whustel
   </fieldset>
</p>

<p align="center">
   <a href="#fifth">[Contact Us]</a>   
   <img alt="Penguin" src="xspenguin.gif" width="11"
   height="19"/>   
   <a href="#first">[Go to Start]</a> 
</p>
</card>
</wml>
```